

Guía básica para el uso de Leftraru



Arquitectura Leftraru



- Infiniband FDR 56Gbps
- Red iLO para adm. hw.
- Red servicio 1 Gbps
- +200 Tb almacenamiento DDN Lustre
- 128 nodos slims
- 4 nodos fats
- 12 Xeon Phi
- Racks enfriados por agua
- Enfriamiento in-row respaldo
- UPS 120 KVA autonomía: 30 mins.



Nodos de cómputo Leftraru

2 tipos de nodo de cómputo:

Slim



Nodo Slim

- 2 CPUS Xeon E5-2660 v2, 10 cores c.u.
- 48 GB RAM DIMM DDR3
- HD Interno 300 GB
- InfiniBand FDR 56 Gbps



Nodo Fat

- 2 CPUS Xeon E5-2660 v2, 10 cores c.u.
- 64 GB RAM DIMM DDR3
- HD Interno 300 GB
- InfiniBand FDR 56 Gbps
- 3 coprocesadores Xeon Phi
 - 240 cores c.u.
 - **8 GB RAM c.u.**



Almacenamiento Leftraru



DDN EXAScaler

- Almacenamiento paralelo de clase mundial
- Alto rendimiento en operaciones IO
- Tolerante a fallas (alta disponibilidad)
- Interconexión infiniband
- Capacidades Big Data

Características en Leftraru

- Sistema archivos EXAScaler (Lustre)
- +200 TB de almacenamiento
- Almacenamiento metadata separado
- 2 controladoras SFA en H.A.
- 4 nodos OSS conectados a Infiniband
- 2 nodos MDS en H.A.



Introducción a SLURM

Simple Linux Utility for Resource Management

- Administra clusters y ejecución de tareas
- Open source
- Utilizado en el 60% de los supercomputadores del top500
- Versión actual instalada en Leftraru: 16.05.4





Características de SLURM

- Tres funciones principales:
 - Asignación de recursos (exclusivos y no exclusivos)
 - Framework para iniciar, ejecutar y monitorear trabajos
 - Gestiona tareas manejando una cola de recursos
- Integra una base de datos para reportes históricos
- Puede reservar diferentes recursos: CPU, socket, nodo o incluso por RAM
 - 6 nodos o 120 CPUs en el caso de Leftraru
- Permite a los administradores modificar tareas en ejecución y realizar reservas



Slurm: Primeros pasos

Pruebas desde la consola:

[usuario@leftraru1 ~]\$ srun -n 2 hostname







Slurm: Monitorear mis tareas desde consola

Monitorear desde la consola:

```
[usuario@leftraru1 ~]$ squeue
```

JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)

0:00

1

cn042

4400799 slims example usuario R

Obtener el nombre de el/los nodos y visitar:

http://monitor.nlhpc.cl/ganglia





Slurm: Monitorear mis tareas desde la web





Slurm: Monitorear mis tareas desde la web



National Laboratory for High Performance Computing Chile

Slurm: Monitorear mis tareas desde la web







Monitorear mis tareas

Puede ingresar a través de ssh a un nodo en donde tenga una tarea en ejecución y ejecutar **htop**

1 2 3 4 5 Mem			100 0 100 0	.0%] .0%] .0%] .0%]	6 [7 [8 [9 [10 [.566	0.0% 0.0% 0.0% 0.7% 0.0%	11 [0. 12 [0. 13 [0. 14 [0. 15 [0. Tasks: 44. 39 thr: 3 r	.0%] 16 [.0%] 17 [.0%] 18 [.0%] 19 [.0%] 20 [running	0.0%] 0.0%] 0.0%] 0.0%] 0.0%]
Swp							θΚ,	/62.5G	Load average: 2.00 2.0	91 2.05	
									optime: 4 days, 00:30:	:11	
PID	USER	PRI	NI	VIRT	RES	SHR S	CPU%	MEM%	IME+ Command		
18305	nperinet	20	0	19020	3768	892 R	100.	0.0	52:34 ./LL_RK4.x		
18335	nperinet	20	0	1/104	1584	892 R	100.	0.0	47:30 ./LL_RK4.X		
10002	root	20	0	105M	2432	14/0 K	0.0	0.0	11 06 /usr/lib/systemd/	(systemd switched r	at sustan da
660	root	20	0	112M	2008	2384 S	0.0	0.0	11.00 /usr/tib/systemu/	systemaswitchea-ro	ootsystemde
671	root	20	0	36816	7236	1004 5 6008 S	0.0	0.0	00.00 /Din/Dash 00.78 /usr/lib/svstemd/	/systemd_iournald	
726	root	20	0	13808	2/28	1268 5	0.0	0.0	AA 83 /usr/lib/systemd/	/systemd-udevd	
1012	root	16		51188	1620	1236 S	0.0	0.0	90.05 /sbin/auditd -n	systema adeva	
1002	root	16		51188	1620	1236 S	0.0	0.0	90.25 /sbin/auditd -n		
1206	root	20	Θ	448M	10956	6968 S	0.0	0.0	00.00 /usr/sbin/Network	kManagerno-daemon	
1209	root	20	O	448M	10956	6968 S	0.0	0.0	00.09 /usr/sbin/Network	kManagerno-daemon	
1139	root	20		448M	10956	6968 S	0.0	0.0	02.34 /usr/sbin/Network	kManagerno-daemon	
1144		20		30220	1564	1300 S	0.0	Θ.Θ	00.44 avahi-daemon: run	nning [cnf004.local]	
1148	dbus	20		28824	1772	1352 S	0.0	0.0	90.44 /bin/dbus-daemon	systemaddress=sy	/stemd:nofork
1166	root	20			1236	776 S	0.0	0.0	00.00 /usr/sbin/gssprox	xy -D	
1167	root	20		198M	1236	776 S	0.0	0.0	00.00 /usr/sbin/gssprox	xy -D	
1168	root	20		198M	1236	776 S	0.0	0.0	00.00 /usr/sbin/gssprox	xy -D	
1169	root	20		198M	1236	776 S	0.0	Θ.Θ	00.00 /usr/sbin/gssprox	xy -D	
1170	root	20	0	198M	1236	776 S	0.0	0.0	00.00 /usr/sbin/gssprox	xy -D	
1164 F1 <mark>Help</mark>	root F2 <mark>Setup</mark>	20 F3 <mark>Se</mark>	oarch	198M F4 F1U	1236 ter <mark>F5</mark> Tr	776 S ee F6 <mark>Sc</mark>	0.0 ortBy	0.0 7 <mark>Nice</mark>	00.30 /usr/sbin/gssprox B <mark>Nice +</mark> F9 <mark>Kill F10Quit</mark>	xy -D	



Slurm: Scripts sbatch

Permiten:

- Ejecutar un scripts batch sin necesidad de estar siempre conectado
- Monitorear el estado de la tarea
- Monitorear recursos
- Monitorear estado de la cola
- Monitorear estado de las particiones



Módulos

Leftraru carga sus programas mediante "módulos"

- Permite varias versiones del mismo software
- No genera conflictos entre versiones
- El software está centralizado

Para cargar un módulo:

module load intel

module avail: muestra todos los nodos disponiblesmodule list: lista todos los módulos cargadosmodule unload: quita un módulo previamente cargado



SBATH Ejemplo de script básico test.sh

Utilizar su editor por consola preferido: vim, nano

#!/bin/bash
#SBATCH --job-name=ejemplo
#SBATCH --partition=slims
#SBATCH -n 1
#SBATCH --output=archivo_%j.out
#SBATCH --error=archivo_%j.err
#SBATCH --mail-user=usuario@gmail.com
#SBATCH --mail-type=ALL
module load intel

1 10

sleep 10

Ejecución: sbatch test.sh



SBATCH Job Array

Slurm permite enviar y administrar millones de trabajos similares de una sola vez.

- Todos los trabajos deben tener las mismas condiciones iniciales
- Provee de variables para controlar la ejecución de los jobs

#!/bin/bash
#SBATCH --job-name=sleep-test
#SBATCH --partition=slims
#SBATCH -n 1
#SBATCH --output=st_%j.out
#SBATCH --error=st_%j.err
#SBATCH --array=1-10
#SBATCH --mail-user=usuario@dominio.cl
#SBATCH --mail-type=ALL
./ejecutable entrada_\$SLURM_ARRAY_TASK_ID



SBATCH Ejemplo OpenMP

#!/bin/bash
#SBATCH --job-name=ejemplo
#SBATCH --partition=slims
#SBATCH --ntasks=1 # igual a parámetro "-n"
#SBATCH --cpus-per-task 20 # "-c"
#SBATCH --output=archivo_%j.out
#SBATCH --error=archivo_%j.err
#SBATCH --mail-user=usuario@gmail.com
#SBATCH --mail-type=ALL

export OMP_NUM_THREADS=20

./ejecutable

Ejecución: sbatch test.sh



SBATCH Intel MPI

#!/bin/bash
#SBATCH --job-name=ejemplo
#SBATCH --partition=slims
#SBATCH -n 2
#SBATCH --output=ejemplo_impi_%j.out
#SBATCH --error=ejemplo_impi_%j.err
#SBATCH --mail-user=usuario@gmail.com
#SBATCH --mail-type=ALL
module load intel impi
srun ./hola_mundo

Hello from thread 00 out of 1 from process 00 out of 2 on cn005 Hello from thread 00 out of 1 from process 01 out of 2 on cn005



Límites generales cuentas de usuario

- 120 CPUs
- 80 GB almacenamiento Lustre
- Walltime 3 días

Solicitud de cuentas:

visitar: http://www.nlhpc.cl (Servicios > Servicios para la Academia > Formulario)

ó solicitar información a info@nlhpc.cl



SSH: Secure Shell - Introducción

- Protocolo seguro de acceso remoto a máquinas
- El cliente también se llama SSH
- Permite:
 - Ejecutar órdenes
 - Redirigir las "X" (sistema ventanas Unix, Linux)
 - Transferir archivos mediante FTP cifrado (sftp)
 - Transferencia de archivos bidireccional (scp)
 - Túneles de conexión, entre otras cosas
- Funciona comúnmente en el puerto TCP 22
- Admite múltiples tipos de cifrado



Funcionamiento Protocolo ssh

- 1. Se determina identidad de cliente y servidor
- 2. Establecimiento de canal seguro (cifrado 256 bits)
- 3. Cliente inicia sesión (autenticación) en el servidor

Dos métodos de autenticación:

- Por clave: mediante credenciales (usuario y password)
- Por llave: el cliente instala su llave pública en el servidor



Instalación ssh

- Linux y Mac OSX: clientes integrados
- Windows:
 - Putty: cliente SSH: <u>http://www.putty.org</u>
 - WinSCP: transferencia de archivos: <u>http://winscp.net</u>



Iniciar Sesión ssh en leftraru

Linux, OSX: ssh USER@leftraru.nlhpc.cl

samuel@leumas_nlbnc'~\$ ssh]	eumas@leftraru_nlbnc_cl
leumas@leftraru_nlbnc_cl's_r	assword.
last login: Eri Aug 12 17:01	:52 2016 from nc520300 in-150-60-117 no
000000000000000000000000000000000000000	
000000000000000 10000000000	00000@0000000000 0000000000000000000000
QQQQQQQQQQQQ @QQQQQQQQQQQQQQQQQQQQQQQQ	000000#0@00000000\$ 000000000000000000
QQQQQQQQR /kQQQQQQQQQQQQQQQQQQQ	0000000100@0000000000 00000000000000000
QQQQQQR "@Q#QRRR@QQQQQQQQQQ	0000@RQ Q#R@QQQQQQQQQQ QQQQQQQQQQQQQQQQQQQQQ
000000@000 , yy0000000000000	000000@00004000R000000 0000000000000
QQQQQQ"RR@QQQQQ RQQQQQQQQQ	000000000000000000000000000000000000000
QQQQQQ ``"RWRQQQQQQQQ	
0000000 `"RR000	00000000000000000000000000000000000000
000000h `@0000vv.	** RWRR000000000000000~ 000000000000000000
00000R RR#000000000	100vv "RBR000000 0000000000000
00000 vvv W @00000000	
<u> а</u> ааа аас	μασασάσσασα μα τη
	ασασσασσασσασσασασσασσασσασσασασα
QQR ; }@Q#@R@C	000000000000000000000000000000000000000
QQ Q @RR,RQQQO	0000000000#0000000000000000000000000000
QQy']Q""^ "[@0	199999999999999999999999999999999999999
QQQQQ \$~RQ0	000000000000000000000000000000000000000
QQQQQQ~ % S@C	000000000000000000000000000000000000000
000000, ,00	0000000 ,@00^@0000000000@RR0@00100 0
000000 000000	R0@ R %0000W#@00000`0000 @0]0^ 0
0000000 :v` B`@#`	V KW040@000000 R00WR'R.000
000000000000R	[@0000"VWR]W #0000
000000000000vvv00000#RR+	~ E01^ 10000000
000000000000000000000000000000000000000	BB@0000000
000000000000000000000000000000000000000	#00000v B00000
000000000000000000000000000000000000000	V0000000000000 @0000
000000000000000000000000000000000000000	, yqqqqqqqqqqqqqqqqqqqq
000000000000000000000000000000000000000	,#עעעעעעעעעעעעעעעעעעע
QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ	********* DTENUENTRO & *********
αάάαααααάαααααάαααααάααα	SIENVENID@ A *********
dddddddddddddddddddddd	LEFIRARU *********
000000000000000000000000000000000000000	
Laboratorio Nacional de Comp	outacion de Alto Rendimiento (NLHPC)
Centro de Modelamiento Matem	natico (CMM)
Universidad de Chile	
IMPORTANTE: NO EJECUTAR PROC	ESOS EN ESTE NODO POR T>10Min
PARA ESTO DEBEN DE USARSE LA	S COLAS DE EJECUCION
[leumas@leftraru2 ~l\$ ■	

Windows:

Cate

Session	Basic options for your PuTTY session					
Logging Terminal Keyboard Bell Features Window	Specify the destination you want to o Host <u>N</u> ame (or IP address) [leftraru.nihpc.cl] Connection type: () Ra <u>w</u> () <u>T</u> elnet() Rlogin()	connect to <u>P</u> ort 22 SSH O Serja				
Appearance Behaviour Translation Selection	Load, save or delete a stored session Sav <u>e</u> d Sessions					
Colours Data Proxy Telnet Rlogin	Default Settings .61	Load Sa <u>v</u> e Delete				
i⊞- SSH	Close window on e <u>xi</u> t: Always Never Only	y on clean exit				



Consideraciones antes de conectar

• Balanceo de carga ssh:

El usuario siempre hará login al nodo menos utilizado (OJO: uso de screen)

- Límite de trabajos en nodos login: 10 minutos
 - Esto afecta también a las transferencias de archivos grandes



ssh: copia de archivos desde línea de comandos

ssh permite copiar desde y hacia equipos remotos

scp archivo.txt <u>usuario@leftraru.nlhpc.cl</u>:~/ copia desde local hacia el servidor

scp <u>usuario@leftraru.nlhpc.cl</u>:~/archivo.txt ~/ copia desde el servidor hacia carpeta local

scp -C archivo.txt usuario@leftraru.nlhpc.cl:~/ "-C" utiliza compresión, envío más rápido

scp -r Directorio <u>usuario@leftraru.nlhpc.cl</u>:~/ copia un directorio completo

Nota: ~/ ruta relativa, indica la raíz del home del usuario



ssh copia de archivos GUI



Filezilla: http://filezilla-project.org

Otras opciones:

- WinSCP
- Midnight Commander (mc)





bash: screen

Cada vez que cerramos sesión, todos los procesos abiertos se cierran. Incluso si los procesos fueron puestos en background.

Screen "desacopla" una sesión completa y la envía al fondo, lo que permite volver a retomarla incluso si la conexión se perdió.

Sólo debe ejecutar screen y luego ejecutar lo que se requiera. Para desacoplar sólo debe presionar la siguiente secuencia: control + a y luego d

Para listar las sesiones abiertas: screen -list Para remontar una sesión: screen -r **id**, donde id es el id de la sesión

Nota: leftraru cuenta con 4 nodos de login, por lo que usted debe recordar donde ejecutó screen



Fin Introducción: Manos a la obra

¿DUDAS?



Ejercicios básicos:

Ejecute el siguiente comando: cp -r /home/courses/ejemplos/ .

- 1.- Ejecute:
 - srun --reservation=curso -N 1 hostname
 - srun --reservation=curso -N 2 hostname
 - srun --reservation=curso -N 3 hostname
- 2.-
 - Cree un script bash que ocupe 40 cores y ejecute el comando sleep 1000
 - Lance el script
 - Intente lanzarlo nuevamente ¿qué sucede?
 - scancel: cancelar tareas
- 3.- Cree un script que reserve 1 nodo completo de forma exclusiva y ejecute el comando sleep 1000
- 4.- Cree un script que lance 4 trabajos, pero sólo dos por cada nodo
- 5.- Cree un script que lance un trabajo de un proceso en un nodo "fat"

Un amigo fiel: man <- sistema de ayuda en los programas linux man sbatch



Ejercicio 6: job array

En el siguiente ejercicio usted jugará con la precisión de cálculo del número pi Ingrese el directorio ejemplos/ejercicio_6, cree el siguiente script y luego ejecútelo:

```
#!/bin/bash
#SBATCH --job-name=pi-test
#SBATCH --partition=slims
#SBATCH -n 1
#SBATCH --output=st_%j.out
#SBATCH --error=st_%j.err
#SBATCH --array=1-10
#SBATCH --mail-user=usuario@mail.cl
#SBATCH --mail-type=ALL
#SBATCH --reservation=curso
module load intel impi
PRECISION=$( echo "$SLURM_ARRAY_TASK_ID*100000000" | bc )
srun ./pi_mpi.exe $PRECISION
```

Vigile la salida: watch -n 1 squeue Una vez terminado los procesos, analice los archivos de salida (cat *.out) ¿Nota alguna diferencia?



Ejercicio 7: Intel - MPI

- 1.- Ingrese a la carpeta ejemplos/ejercicio_7
- 2.- Compile el programa hello.c con intel mpi mpiicc hello.c -o hello -fopenmp Ejecútelo con srun -n 1 Ejecútelo con srun -n 20 Ejecútelo con srun -n 21 Ejecútelo con srun -c 1 Ejecútelo con srun -c 2 Ejecútelo con srun -c 20 Ejecútelo con srun -c 21 (¿qué ocurre?) Ejecútelo con srun -n 1 -c 20 Ejecútelo con srun -n 2 -c 20



Ejercicio 8: Intel - MPI

- 1.- Ingrese a la carpeta ejemplos/ejercicio_8
- 2.- Compile el programa hello.c con intel mpi mpiicc hello.c -o hello -fopenmp
- 3.- Ejecute la aplicación con 2 procesos MPI y 20 OpenMP
- 4.- Analice los archivos de salida



Ejercicio 9: SBATCH Control tareas por RAM

- Ingrese al directorio ~/ejemplos/ejercicio_9
- Vea el contenido el ejemplo script.sh (cat script.sh)
- Ejecute y vigile la tarea
 - ¿Cuántos nodos ocupa?
 - ¿Cuanta RAM ocupa?
 - Cancele la tarea
- Edite nuevamente script_slurm.sh
 - Añada la línea: #SBATCH --mem-per-cpu=8192
- Vuelva a lanzar script_slurm.sh
 - ¿Qué ocurre?
 - ¿Cuanta RAM ocupa?
 - ¿Cuántos nodos ocupa ahora la tarea?



Ejercicio 10: Compilación - mkl

Intel MKL es una librería de optimización matemática que utiliza vectorización para mejorar el rendimiento

- 1.- Ingrese a la carpeta ejemplos/ejercicio_10 y liste los archivos (ls -l)
- 2.- Ingrese a matrix/linux
 - El makefile de este ejemplo puede producir dos binarios: icc y gcc
 - Para producir el binario con gcc: make gcc, producirá un archivo matrix.gcc
 - Para producir el binario con icc: make gcc, producirá un archivo matrix.icc
 - Ejecute ambos binarios
 - ./matrix.gcc
 - ./matrix.icc
- 3.- Retroceda a ingrese al directorio matrix/linux_mkl (cd ../linux_mkl)
 - El makefile de este ejemplo puede generar un binario con intel MKL
 - Para producir el binario con mkl: make mkl, producirá un archivo matrix.mkl
 - OMP_NUM_THREADS es una variable que define la cantidad de procesos openmp
 - export OMP_NUM_THREADS=1
 - ./matrix.mkl



Ejercicio 10: Compilación - mkl

Ejercicio práctico:

- 1. Cree un script sbatch que ejecute el binario matrix.mlk, utilizando 1 proceso y 20 threads openmp
- 2. Modifique su script para lanzar ahora 2 procesos y 20 threads openmp

Hilos openmp: export OMP_NUM_THREADS=XX donde XX es el número de hilos

Compare los resultados



Ejercicio 11: Reservas mal hechas

- Ingrese a la carpeta ~/ejemplos/ejercicio_11
- Analice el archivo script.sh
 - ¿Cuántas CPU está reservando?
- Ejecútelo
- Vigile su tarea con ganglia
 - ¿Cuántas CPU está utilizando?
 - Cancele la tarea
- Modifique su script para que utilice las CPU que corresponden



Compilación y Optimización

- Ingrese a la carpeta "ejemplos" de su home
- Cargue el módulo intel
- Compile el ejemplo pi.c sin ningún flag de optimización y ejecútelo
- Compile el ejemplo pi.c con el flag "-O3" y ejecútelo
 - ¿Cuál fue la diferencia?
- Compile el ejemplo pi_openmpi.c
 - Ejecútelo con 1 sólo proceso
 - Ejecútelo con 10 procesos
 - Ejecútelo con 60 procesos
 - ¿Cuál es la diferencia?





¿DUDAS?

soporte@nlhpc.cl

